UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/693,749	10/24/2003	Thekkthalackal Varugis Kurien	MVIR-0110 / 301118.01	2449

41505 7590 05/11/2009

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
LE, CANH	

ART UNIT	PAPER NUMBER
2439	

MAIL DATE	DELIVERY MODE
05/11/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/693,749	KURIEN ET AL.	
	Examiner	Art Unit	
	CANH LE	2439	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 February 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25 and 35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☒ Claim(s) *13,20 and 25* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 02/27/2009 has been entered.

This Office Action is in response to the communication filed on 02/27/2009.

Claims 26-34 have been cancelled.

Claims 1, 13, 16, 18 and 25 have been amended.

Claim 35 have been added.

Claims 1-25 and 35 have been examined and are pending.

Response to Arguments

The Applicant's arguments with respect to claims 1-25 and 35 have been considered but are moot in view of the new ground(s) of rejection.

Claim Objections

Claims 13, 20, and 25 are objected to because of the following informalities: Appropriate correction is required.

(Claim 13, line 15): ", resistance" should replace ", a resistance ".

(Claim 20, line 1): "said policy" should replace "said assurance policy" to avoid potentially antecedent basis.

(Claim 25, line 31): "the first environment" should replace "the first software environment" to avoid potentially antecedent basis.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-22 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claim 1 recites "applies the corresponding ***wrapper*** to the first of the plurality of data, the ***corresponding wrapper*** comprising ***the second software object and seal*** that may be checked" in lines 28-30 and recites "applies ***a second wrapper*** to the processed data, ***the second wrapper*** comprising ***the first software object*** and ***a second seal*** that may be checked" in lines 35-36" (emphasis added).

However, the specification does not enable one having ordinary skill in the art to make and/or use an invention that creates a wrapper for data, when the wrapper includes a software

object (which, as defined by the claim, is an executing software application). It is not described how an executing software application can be included in a wrapper.

Claims 2-12 are dependent on claim 1, and therefore inherit the 35 U.S.C 112, first paragraph as failing to comply with the enablement requirement of the independent claims.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-12, 23-24 and 25 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites the limitation "sending said first of the plurality of data to ***said base environment***" in line 14 (emphasis added). It is unclear as to whether "said base environment" refers to *"said base component"* or other "base environment".

Claim 1 recites the limitation "unmodified by comparing said data to the corresponding wrapper, said ***second software*** processing" in lines 21-22 (emphasis added). It is unclear as to whether "said second software" refer to "second software object" or "second software environment".

Claim 1 recites the limitation "the operation" in line 6. There is insufficient antecedent basis for this limitation in the claim.

Claim 3 recites the limitation "the content" in line 3. There is insufficient antecedent basis for this limitation in the claim.

Claim 4 recites the limitation "the resistance" in line 1 and limitation "the display" in line 3. There is insufficient antecedent basis for this limitation in the claim.

Claim 5 recites the limitation "the resistant" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 7 recites the limitation "the signature" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 10 recites the limitation "the behavior" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 13 recites the limitation "the second software environment" in line 10. There is insufficient antecedent basis for this limitation in the claim.

Claim 13 recites the limitation "the result" in line 18. There is insufficient antecedent basis for this limitation in the claim.

Claim 17 recites the limitation "the content" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 19 recites the limitation "the input" in line 3. There is insufficient antecedent basis for this limitation in the claim.

Claim 23 recites the limitation "the behavior" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 23 recites the limitation "said specification" in line 4. It is unclear "said specification" as to whether refer to "first specification" or "second specification".

Claim 24 recites the limitation "the behavior" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 25 recites the limitation "said first software object" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 25 recites the limitation "the second software environment" in lines 10-11. There is insufficient antecedent basis for this limitation in the claim.

Claim 25 recites the limitation "the result" in line 18. There is insufficient antecedent basis for this limitation in the claim.

Claim 25 recites the limitation "the first software environment" in line 21. There is insufficient antecedent basis for this limitation in the claim.

Claim 25 recites the limitation "the second computing environment" and "the first computing environment" in lines 31-32. There is insufficient antecedent basis for this limitation in the claim.

Claims 2-12 are dependent on claim 1, and therefore inherit the 35 U.S.C 112, and therefore inherit the 35 U.S.C 112, second paragraph as failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The Examiner kindly requests the Applicant to point out with specificity (i.e. column and line) in the specification where it describes/supports the above limitation.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 13-24 and 35 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject mater.

Regarding to claim 13, the claim invention is not directed to eligible subject matter under 35 U.S.C § 101 in view of ***In re Bilski,*** 88 USPQ 2d 1385 CAFC (2008). While the claims recite a series of steps or acts to be performed, a statutory "process" under 35 U.S.C § 101 process must (1) be tied to a particular machine or (2) transform underlying subject matter (such as an article or materials) to a different state or thing (See ***In re Bilski,*** 88 USPQ 2d 1385 CAFC (2008); see also ***Diamond v. Diehr,*** 450 U.S. 175, 184 (1981)***; Parker v. Flook,*** 437 U.S. 584, 588 n.9 (1978)***; Gottschalk v. Benson,*** 409 U.S. 63, 70 (1972)*; Cochrane v. Deener,* **94** U.S. 780,787-88 (1876)); The instant claims are neither positively tied to a particular machine that accomplishes the claimed method steps nor transform underlying subject matter. The method claimed including steps of "determining, by the first software object", "sending, by the first software object", "determining, by the base environment", "applying, by the base environment", "sending, by the base environment", "creating, by the second software environment", processing, by the second software environment", "sending, by the second software environment", "determining, by the base environment", "applying/sending, by the base environment", and "verifying, by the first software environment" is broad enough that the claim could be completely performed mentally, verbally or without a machine nor is any transformation apparent; Therefore, the claimed invention is directed to non-statutory subject matter.

Claims 14-24 and 35 are rejected with the same reason above.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 5-14, 19-20, 23-25, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Muschellack et al.** (US 7,309,004 B1) in view of **Tal Garfinkel et al.,** *"Terra: A virtual Machine-Based Platform for Trusted Computing", pages 1-14, October 19-22, 2003.*

As per claim 1:

Muschellack teaches a system that manages the partitioning of an application comprising:

(a) at least one processor and at least one memory in communication with said at least one processor, said processor configured to execute program instructions **[Muschellack: fig. 8]** that comprise the following:

(b) a base component stored in said at least one memory that hosts the operation of a first environment and a second environment application **[Muschellack : Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash,**

deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"; See also Col. 20, lines 29-40; "With a sealed storage, the trust ATM component has access to secret information stored in the sealed storage in a data store of the ATM which is not available to other software is trusted by the trusted ATM component"] comprising:

(c) a first software object of said application that executes in said first environment comprising a first operating system, wherein said first software object provides a subset of the operations of the application; said first software object handling a plurality of data and including logic to identify a first of said plurality of data as not processable by said first software object, **[Muschellack : fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430; Col. 19, lines 17-27; "the terminal control software components 422 of the ATM may be granted permission to operate in the nexus mode. In a TP based on Microsoft's NGSCB specification such components may be programmed to use features of the TPM through communication with the nexus 420…"]** *[[said first software object sending said first of said plurality of data to said base environment, said first software object receiving processed data corresponding to said first of said plurality of data from said base environment, said first software object using said processed data to further process the plurality of data]];* and

(d) a second software object of said application that executes in said second environment comprising a second operating system **[Muschellack : fig. 8; Nexus mode, Nexus; Col. 19, line 3-7, the nexus mode (i.e. right hand side 432); Col. 19, lines 17-27; "the terminal control software components 422 of the ATM may be granted permission to operate in the nexus**

mode. In a TP based on Microsoft's NGSCB specification such components may be programmed to use features of the TPM through communication with the nexus 420…"], *[[said second software object receiving said first of said plurality of data from said base component with a corresponding wrapper, said second software object verifying said first of said plurality of data as being unmodified by comparing said data to the corresponding wrapper, said second software processing said first of said plurality of data in a manner that resists tampering with said first of said plurality of data, said second software object sending said processed data to said base component]];*

(c) said base component comprising or hosting logic that receives said first of said plurality of data from said first software object **[Muschellack : fig. 8; Col. 20; line 17-28; See also Col. 20, lines 29-40]** *[[applies the corresponding wrapper to the first of said plurality of data, said corresponding wrapper comprising said second software object and a seal that may be checked against said first of said plurality of data to determine whether said first of said plurality of data has been altered since the seal was determined, and routes said first of said plurality of data to said second environment, such that functionality of said application is parsed between said first and second operating systems.]];*

Muschellack does not clearly disclose in details,

(b1) a base component stored in said at least one memory that hosts the operation of a first environment and a second environment application;

(c1) said first software object sending said first of said plurality of data to said base environment, said first software object receiving processed data corresponding to said first of

said plurality of data from said base environment, said first software object using said processed data to further process the plurality of data;

(d1) said second software object receiving said first of said plurality of data from said base component with a corresponding wrapper, said second software object verifying said first of said plurality of data as being unmodified by comparing said data to the corresponding wrapper, said second software processing said first of said plurality of data in a manner that resists tampering with said first of said plurality of data, said second software object sending said processed data to said base component;

(e1) said base component comprising or hosting logic that receives said first of said plurality of data from said first software object, applies the corresponding wrapper to said first of said plurality of data, said corresponding wrapper comprising said second software object and a seal that may be checked against said first of said plurality of data to determine whether said first of said plurality of data has been altered since the seal was determined, and routes said first of said plurality of data to said second environment, such that functionality of said application is parsed between said first and second operating systems;

(f) said base component further comprising or hosting logic that receives said processed data from said second software object, applies a second wrapper to said processed data, said second wrapper comprising said first software object and a second seal that may be checked against said processed data to determine whether said processed data has been altered since the second seal was determined, and routes said processed data to said first environment, such that functionality of said application is parsed between said first and second operating systems;

However, Tal discloses a virtual machine-based platform for trusted computing, wherein

(b1) a base component stored in said at least one memory that hosts the operation of a first environment and a second environment application **[Tal: pg. 2; section 2. Terra Architecture; Trusted virtual machine monitor (TVMM), Open-box run commodity operating system and Closed-box VM run a special trusted OS with a selection of applications, providing similar to the NGSCB; See also 2.1 The Trusted Virtual Machine Monitor; fig. 1]**;

(c1) said first software object sending said first of said plurality of data to said base environment, said first software object receiving processed data corresponding to said first of said plurality of data from said base environment, said first software object using said processed data to further process the plurality of data **[Tal: pg. 2; section 2. Terra Architecture; Trusted virtual machine monitor (TVMM), Open-box run commodity operating system and Closed-box VM run a special trusted OS with a selection of applications, providing similar to the NGSCB; See also 2.1 The Trusted Virtual Machine Monitor; fig. 1]**;

(d1) said second software object receiving said first of said plurality of data from said base component with a corresponding wrapper, said second software object verifying said first of said plurality of data as being unmodified by comparing said data to the corresponding wrapper, said second software processing said first of said plurality of data in a manner that resists tampering with said first of said plurality of data, said second software object sending said processed data to said base component **[Tal: pg. 5, section 3.1 Local Security Model; TVMM runs at highest privilege level. It is "root secure", meaning that it is secure from tampering even by the platform owner who has root level access, from the management VM; pg. 3; section 2.1 The Trusted Virtual Machine Monitor; TVMM provides traditional VMMs;**

Multiple applications run in different virtual machines. Isolation - Abstraction of separate physical machines provides an intuitive model for understanding the isolation properties of the platform; Extensibility - configuring applications based on level assurance and resource requirement; Compatibility - The greater isolation of a VM can improve assurance on its own; untrusted applications can be transformed into low-assurance trusted applications in closed boxes with the minimal changes; See also pgs. 8-9, section 4.6 Hardware Support for Trusted VMMs; Seal Storage - Encrypts data under the private key of the temper-resistant coprocessor that is responsible for attestation …The coprocessor will only allow a trusted OS with the same hash that sealed data to unseal it; fig. 1; sealed storage device].

(e) said base component comprising or hosting logic that receives said first of said plurality of data from said first software object, applies the corresponding wrapper to said first of said plurality of data, said corresponding wrapper comprising said second software object and a seal that may be checked against said first of said plurality of data to determine whether said first of said plurality of data has been altered since the seal was determined, and routes said first of said plurality of data to said second environment, such that functionality of said application is parsed between said first and second operating systems **[Tal: pg. 5, section 3.1 Local Security Model; TVMM runs at highest privilege level. It is "root secure", meaning that it is secure from tampering even by the platform owner who has root level access, from the management VM; pg. 3; section 2.1 The Trusted Virtual Machine Monitor; TVMM provides traditional VMMs; Multiple applications run in different virtual machines. Isolation - Abstraction of separate physical machines provides an intuitive model for understanding the isolation properties of the platform; Extensibility - configuring**

applications based on level assurance and resource requirement; Compatibility - The greater isolation of a VM can improve assurance on its own; untrusted applications can be transformed into low-assurance trusted applications in closed boxes with the minimal changes; See also pgs. 8-9, section 4.6 Hardware Support for Trusted VMMs; Seal Storage - Encrypts data under the private key of the temper-resistant coprocessor that is responsible for attestation …The coprocessor will only allow a trusted OS with the same hash that sealed data to unseal it; fig. 1; sealed storage device].

(f) said base component further comprising or hosting logic that receives said processed data from said second software object, applies a second wrapper to said processed data, said second wrapper comprising said first software object and a second seal that may be checked against said processed data to determine whether said processed data has been altered since the second seal was determined, and routes said processed data to said first environment, such that functionality of said application is parsed between said first and second operating systems **[Tal: pg. 5, section 3.1 Local Security Model; TVMM runs at highest privilege level. It is "root secure", meaning that it is secure from tampering even by the platform owner who has root level access, from the management VM; pg. 3; section 2.1 The Trusted Virtual Machine Monitor; TVMM provides traditional VMMs; Multiple applications run in different virtual machines. Isolation - Abstraction of separate physical machines provides an intuitive model for understanding the isolation properties of the platform; Extensibility - configuring applications based on level assurance and resource requirement; Compatibility - The greater isolation of a VM can improve assurance on its own; untrusted applications can be transformed into low-assurance trusted applications in**

closed boxes with the minimal changes; See also pgs. 8-9, section 4.6 Hardware Support for Trusted VMMs; Seal Storage - Encrypts data under the private key of the temper-resistant coprocessor that is responsible for attestation …The coprocessor will only allow a trusted OS with the same hash that sealed data to unseal it; fig. 1; sealed storage device].

Therefore, it would have been obvious to one person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack by including the teaching of Tal to provide a simple and flexible programming model that allows application designers to build secure application in the same way they would on a dedicated closed platform **[Tal: pg. 3, Col. 1, 2nd paragraph].**

As per claim 5:

The combination of Muschellack and Tal teach the subject matter as described above.

Muschellack further teaches said first of said plurality of said is entered on a keyboard **[Muschellack : fig. keyboard 16]**, and wherein the resistant to tampering provided by said second software object comprises resisting tampering with said first of said plurality of data in transit from said keyboard to an input stream of said second software object **[Muschellack :fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"].**

As per claim 6:

Muschellack further teaches the system of claim 5, wherein said second software object signs said first of said plurality of data to prevent subsequent tampering with said first of said plurality of data **[Muschellack : Col. 9, line 60 to Col. 10, line 3; "The components (i.e. application) may include or have access to applications which provide cryptographic functions for performing, encryption, decryption, digital signature signing, digital signature verification, hashing and/or other cryptographic calculation…a secure communication session between components"].**

As per claim 7:

Muschellack further teaches the system of claim 6, wherein said second environment signs said first of said plurality of data and the signature created by said second application as an indication that said first of said plurality of data and said signature were created in said second environment **[Muschellack : Col. 9, line 60 to Col. 10, line 3; "The components (i.e. application) may include or have access to applications which provide cryptographic functions for performing, encryption, decryption, digital signature signing, digital signature verification, hashing and/or other cryptographic calculation…a secure communication session between components"].**

As per claim 8:

The combination of Muschellack and Tal teach the subject matter as described above.

Tal further teaches the system, wherein a base component comprises a component that assigns a first identifier to said second environment **[Tal: fig. 1; pg. 1-3; TVMM run standard virtual machine ("open-box VM") and closed-box virtual machines ("closed-box VMs"). Both open- and closed-box VMs provide a raw hardware interface that is practically identical to the underlying physical machine; TVMM Virtual machine controls different Operating System in different environments; TVMM supports multiple operating systems by assigning different identifiers and software objects in different environments].**

As per claim 9:

The combination of Muschellack and Tal teach the subject matter as described above.

Tal further teaches said first of said plurality of data includes, or is accompanied by, said first identifier and a second identifier that identifies said second software object **[Tal: fig. 1; pg. 1-3; TVMM run standard virtual machine ("open-box VM") and closed-box virtual machines ("closed-box VMs"). Both open- and closed-box VMs provide a raw hardware interface that is practically identical to the underlying physical machine; TVMM Virtual machine controls different Operating System in different environments; TVMM supports multiple operating systems by assigning different identifiers and software objects in different environments].**

As per claim 10:

The combination of Muschellack and Tal teach the subject matter as described above.

Muschellack further teaches said first environment is associated with a first specification that describes the behavior of said first environment, wherein said second environment is associated with a second specification that describes the behavior of said second environment, wherein there is a higher level of assurance that said second environment will conform to said second specification than that said first environment will conform to said first specification **[Muschellack : fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"].**

As per claim 11:

Muschellack further teaches the system of claim 10, wherein said second software object relies upon the behavior of the second environment in order to resist tampering with said first of said plurality of data **[Muschellack : fig. 8, Nexus Mode 432].**

As per claim 12:

The combination of Muschellack and Tal teach the subject matter as described above.

Tal further teaches said base component is said second environment, or is included within said second environment **[Tal: fig. 1, A Trusted virtual machine monitor (TVMM); pgs. 2-3, section 2. Terra Architecture].**

As per claim 13:

Muschellack teaches a method of a first software object of an application, which executes in a first environment comprising a first operating system **[Muschellack : fig. 8; standard mode Operating system]** , handling data to which an assurance policy that corresponds to a level of assurance that the application will perform its expected functions correctly applies, the method comprising:

(a) determining, by the first software object, that the data should be processed securely **[Muschellack : fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430]**;

(b) sending, by the first software object, the data to a base environment with an indication to process the data securely **[Muschellack : fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430; Col. 19, lines 17-27; "the terminal control software components 422 of the ATM may be granted permission to operate in the nexus mode. In a TP based on Microsoft's NGSCB specification such components may be programmed to use features of the TPM through communication with the nexus 420…"]**;

(f) creating, by the second software environment, resistance to tampering **[Muschellack: fig. 8; Nexus mode 414; Col. 19, lines 10-17]** *[[comprising determining that the data has not been modified using the data and the corresponding seal]]*;

(g) processing, by the second software environment, the data **[Muschellack: fig. 8; Nexus mode 414; Col. 19, lines 10-17]**;

(h) sending, by the second software environment, the result of the processed data to the base environment with an indication to return the data to a software environment that originally

sent the data **[Muschellack : Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"];**

Muschellack does not clearly disclose in details wherein,

(c) determining, by the base environment, a second software object with which to process the data securely, the second software object corresponding to the first software object;

(d) applying, by the base environment, a wrapper to the data, the wrapper identifying the second software environment, the wrapper comprising a seal that may be checked against the data to determine whether the data has been altered since the seal was determined;

(e) sending, by the base environment, the data and the corresponding wrapper to the second software environment;

(l) determining, by the base environment, that the first software environment is the software environment that originally sent the result;

(m) applying, by the base environment, a second wrapper to the result, the second wrapper identifying the first software environment, the second wrapper comprising a second seal that may be checked against the result to determine whether the result has been altered since the second seal was determined;

(n) sending, by the base environment, the result and the second wrapper to the first software environment; and

(o) verifying, by the first software environment, that the result has not been modified using the result and the second seal.

However, Tal discloses a virtual machine-based platform for trusted computing wherein,

(c-e) determining, by the base environment, a second software object with which to process the data securely, the second software object corresponding to the first software object; applying, by the base environment, a wrapper to the data, the wrapper identifying the second software environment, the wrapper comprising a seal that may be checked against the data to determine whether the data has been altered since the seal was determined; sending, by the base environment, the data and the corresponding wrapper to the second software environment [**Tal: pg. 3; section 2.1 The Trusted Virtual Machine Monitor; TVMM provides traditional VMMs; Multiple applications run in different virtual machines. Isolation - Abstraction of separate physical machines provides an intuitive model for understanding the isolation properties of the platform; Extensibility - configuring applications based on level assurance and resource requirement; Compatibility - The greater isolation of a VM can improve assurance on its own; untrusted applications can be transformed into low-assurance trusted applications in closed boxes with the minimal changes; See also pgs. 8-9, section 4.6 Hardware Support for Trusted VMMs; Seal Storage - Encrypts data under the private key of the temper-resistant coprocessor that is responsible for attestation …The coprocessor will only allow a trusted OS with the same hash that sealed data to unseal it; fig. 1; sealed storage device**];

(l-n) determining, by the base environment, that the first software environment is the software environment that originally sent the result; applying, by the base environment, a second

wrapper to the result, the second wrapper identifying the first software environment, the second wrapper comprising a second seal that may be checked against the result to determine whether the result has been altered since the second seal was determined; sending, by the base environment, the result and the second wrapper to the first software environment; determining that the data has not been modified using the data and the corresponding seal [**Tal: pg. 3; section 2.1 The Trusted Virtual Machine Monitor; TVMM provides traditional VMMs; Multiple applications run in different virtual machines. Isolation - Abstraction of separate physical machines provides an intuitive model for understanding the isolation properties of the platform; Extensibility - configuring applications based on level assurance and resource requirement; Compatibility - The greater isolation of a VM can improve assurance on its own; untrusted applications can be transformed into low-assurance trusted applications in closed boxes with the minimal changes; See also pgs. 8-9, section 4.6 Hardware Support for Trusted VMMs; Seal Storage - Encrypts data under the private key of the temper-resistant coprocessor that is responsible for attestation …The coprocessor will only allow a trusted OS with the same hash that sealed data to unseal it; fig. 1; sealed storage device**]; and

(o) verifying, by the first software environment, that the result has not been modified using the result and the second seal [**Tal: pgs. 3-5, section 2.2 Attestation and VM Identity: Attestation enables an application in a VM to authenticate itself to remote parties ...The party receiving an attestation must judge for itself how strongly it believes in the correctness and security of each of the platform's layers ...Attestation requires building a**

certificate chain, from the tamper-resistant hardware all the way to an application VM, to identity each component of the software stack..].

Therefore, it would have been obvious to one person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack by including the teaching of Tal to provide a simple and flexible programming model that allows application designers to build secure application in the same way they would on a dedicated closed platform **[Tal: pg. 3, Col. 1, 2nd paragraph].**

As per claim 14:

Muschellack and Tal teach the method of claim 13.

Muschellack and Tal further teach the method, wherein the resistance to tampering comprises a resistance to a change in said data **[Muschellack : Col. 20, lines 29-46; sealed storage; Tal: pg. 2, section 2. Terra Architecture; fig. 1; Sealed storage device].**

As per claim 19:

This claim has limitations that are similar to those of claim 5, thus it is rejected with the same rationale applied against claims 5 above.

As per claim 20:

The combination of Muschellack and Tal teach the subject matter as described above.

Muschellack and Tal further teach said policy specifies that said data is to be handled by said second software object **[Muschellack : fig. 8; Col. 20; line 17-28; "software components**

(i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"; an application (i.e. second software object) runs on a high-assurance environment (i.e. RHS); pgs. 3-4; 2.2 Attestation and VM Identity section].

As per claim 23:

The combination of Muschellack and Tal teach the subject matter as described above.

Muschellack further teaches said second environment is associated with a first specification that describes the behavior of said second environment, and wherein said assurance policy provides that said second environment will conform to said specification **[Muschellack : fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"; an application (i.e. second software object) runs on a high-assurance environment (i.e. RHS)]; A second environment run on the RHS which relates to high-assurance is associated with specification that describe its behavior].**

As per claim 24:

The combination of Muschellack and Tal teach the subject matter as described above.

Muschellack and Tal further teaches said first environment is associated with a second specification that describes the behavior of said first environment, and wherein said first environment provides a second level of assurance that actions performed in the first environment will be performed correctly, said second level of assurance being relatively lower than said first level of assurance [**Muschellack : fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"; an application (i.e. second software object) runs on a high-assurance environment (i.e. RHS); A level of assurance of the standard mode (i.e. LHS) is relatively lower than a level of assurance of the Nexus mode (i.e. RHS); Tal: pgs. 1-2; 1. Introduction & 2. Terra Architecture section: open-box VM & closed-box VMs].**

As per claim 25:

This claim has similar limitation as claim 13 with additional limitations (p) determining, by the first environment, that the second computing environment has a corresponding level of trust sufficient for the first computing environment to trust the result [**Tal: pgs. 3-5, section 2.2**

Attestation and VM Identity: Attestation enables an application in a VM to authenticate itself to remote parties ...The party receiving an attestation must judge for itself how strongly it believes in the correctness and security of each of the platform's layers ...Attestation requires building a certificate chain, from the tamper-resistant hardware all the way to an application VM, to identity each component of the software stack…] ; and (q) using the result to execute the first software object [**Tal: pgs. 3-5, section 2.2 Attestation and VM Identity: Attestation enables an application in a VM to authenticate itself to remote parties ...The party receiving an attestation must judge for itself how strongly it believes in the correctness and security of each of the platform's layers ...Attestation requires building a certificate chain, from the tamper-resistant hardware all the way to an application VM, to identity each component of the software stack…].** Thus is rejected with the same rationale against claim 13 above.

As per claim 35:

The combination of Muschellack and Tal teach the subject matter as described above. Tal further teaches the method of claim 13, further comprising: determining, by the first software environment, that the second software environment has a corresponding level of trust sufficient for the first software environment to trust the result; and using the result to execute the first software object [**Tal: pgs. 3-5, section 2.2 Attestation and VM Identity: Attestation enables an application in a VM to authenticate itself to remote parties ...The party receiving an attestation must judge for itself how strongly it believes in the correctness and security of each of the platform's layers ...Attestation requires building a certificate chain, from the**

tamper-resistant hardware all the way to an application VM, to identity each component of the software stack…] .

Claims 2-4 and 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Muschellack et al.** (US 7,309,004 B1) in view of **Tal Garfinkel et al.,** *"Terra: A virtual Machine-Based Platform for Trusted Computing", pages 1-14, October 19-22, 2003,* further in view of **Clapper** (US 2003/0107584 A1).

As per claim 2:

Muschellack and Tal do not explicitly teach the system of first software object causes a representation of said first of said plurality of data to be displayed on a display device, said representation comprising one or more indecipherable graphics.

However, in an analogous art, Clapper teaches a security system for visual display, wherein data is displayed on a display device, said representation comprising one or more indecipherable graphics **[Clapper: fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display].**

Therefore, it would have been obvious to one person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack and Tal by including the teaching of Clapper, wherein data is displayed on a display device, said representation comprising one or more indecipherable graphics to provide secure viewing of sensitive information on a display **[Clapper; par. [0004], lines 5-6].**

As per claim 3:

Clapper further teaches the system of claim 2, wherein said one or more indecipherable graphics are either:

the same size as each other **[Clapper: fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display].**

As per claim 4:

The combination of Muschellack and Tal teach the subject matter as described above.

Muschellack and Tal do not explicitly teach a system, wherein the resistance to tampering provided by said second software object comprises said second environment resisting interference with the display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed.

However, in an analogous art, Clapper teaches a security system for visual display, wherein the resistance to tampering provided by said second software object comprises said second environment resisting interference with the display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed **[Clapper: fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display; par. [0042]].**

Therefore, it would have been obvious to one person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack and Tal by including the teaching of Clapper, wherein the resistance to tampering provided by said second software object comprises said second environment resisting interference with the display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed to provide secure viewing of sensitive information on a display **[Clapper; par. [0004], lines 5-6].**

As per claim 15:

This claim has limitations that are similar to those of claim 4, thus it is rejected with the same rationale applied against claims 4 above.

As per claim 16:

This claim has limitations that are similar to those of claim 2, thus it is rejected with the same rationale applied against claims 2 above.

As per claim 17:

This claim has limitations that are similar to those of claim 3, thus it is rejected with the same rationale applied against claims 3 above.

As per claim 18:

Clapper further teaches directing, by said first software object or said second software object, or a combination of said first software object and said second software object, a component responsible for visual output to display items displayed on said visual display device to be changed in at least one respect to allow permit viewing of an image of the data produced by said second software object **[Clapper: fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display].**

Claims 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Muschellack et al.** (US 7,309,004 B1) in view of **Tal Garfinkel et al.,** *"Terra: A virtual Machine-Based Platform for Trusted Computing", pages 1-14, October 19-22, 2003,* further in view of **Hayman et al**. (US 5,895,966).

As per claim 21:

Muschellack and Tal do not explicitly teach data includes, or is associated with, a first label that identifies said second environment as a location in which said data is to be processed.

However, in an analogous art, Hayman teaches a security system for computer systems, wherein data includes, or is associated with, a first label that identifies said second environment as a location in which said data is to be processed **[Hayman: abstract, fig. 3A, fig. 3B, col. 1, lines 63-64, col. 5, line 24 to col. 6, line 36; security labels are placed on each data file].**

Therefore, it would have been obvious to one person of ordinary skill in the art at the time the invention was made to combine the method of Muschellack and Tal by including the teaching of Hayman, wherein data includes, or is associated with, a first label that identifies said

second environment as a location in which said data is to be processed in order to provide users with a means for placing security labels on each data file or other system resource, and on each user process to enable to determine who has what type of access to which data file or other system resources **[Hayman, col. 1, line 64 to Col. 2, line 1].**

As per claim 22:

Hayman further teaches said data includes, or is associated with, a second label that identifies said second software object as a processor for said data, and wherein said second environment routes said data to said second software object based on said second label **[Hayman: abstract, fig. 3A, fig. 3B, col. 1, lines 63-64, col. 5, line 24 to col. 6, line 36; security labels are placed on each data file].**

Conclusion

The examiner requests, in response to this Office action, support be shown for language added to any original claims on amendment and any new claims. That is, indicate support for newly added claim language by specifically pointing to page(s) and line number(s) in the specification and/or drawing figure(s). This will assist the examiner in prosecuting the application. Failure to show support can result in a non-compliant response.

When responding to this office action, Applicant is advised that if Applicant traverses an obviousness rejection under 35 U.S.C. 103, a reasoned statement must be included explaining

why the Applicant believes the Office has erred substantively as to the factual findings or the conclusion of obviousness See 37 CFR 1.111(b).

Additionally Applicant is further advised to clearly point out the patentable novelty which he or she thinks the claims present, in view of the state of the art disclosed by the references cited or the objections made. He or she must also show how the amendments avoid such references or objections See 37 CFR 1.111(c).

The prior arts made of record and not relied upon are considered pertinent to applicant's disclosure.

US 7313687 B2 to Kaler; Christopher G. et al.;

US 7228426 B2 to Sinha; Saurabh et al.;

Christian Jensen and Daniel Hagimont, "Protection Wrappers A Simple and Portable Sandbox for Untrusted Applications", ACM SIGOPS European Workshop, pgs. 104-110, 1998.

Paul England and Marcus Peinado, "Authenticated Operation of Open Computing Devices", Springer Berlin / Heidelberg, pages 346-361, January 01, 2002.

Tal Garfinkel , Mendel Rosenblum, and Dan Boneh, "Flexible OS Support and Application for Trusted Computing", pages 1-6, May 18-21, 2003.

Tal Garfinkel , Mendel Rosenblum, and Dan Boneh, "Flexible OS Support and Application for Trusted Computing", pages 1-6, May 18-21, 2003.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Canh Le whose telephone number is 571-270-1380. The examiner can normally be reached on Monday to Friday 7:30AM to 5:00PM other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Zand Kambiz can be reached on 571-272-3811. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Canh Le/

Examiner, Art Unit 2439

May 6, 2009

/Kambiz Zand/
Supervisory Patent Examiner, Art Unit 2434